Instituto Federal do Sul de Minas Gerais

Projeto e Análise de Algoritmos

(Grafos)

Aula 04 – Busca em Largura

Material - Prof. Humberto César Brandão de Oliveira

Prof. Douglas Castilho

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL DE MINAS GERAIS

Últimas aulas

- Aula 01: Introdução:
  - História;
  - Aplicações
- Aula 02: Conceitos Básicos:
  - Grafo simples;
  - Grafo completo/vazio;
  - Grafo não orientado:
    - Arestas laço;
    - Arestas paralelas;
  - Grafo orientado;
  - Grafo valorado;
- Aula 03: Representação Computacional:
  - Matriz de adjacência;
  - Matriz de incidência;
  - Lista de adjacência;
- Aula 04: Busca em Profundidade:
  - Método recursivo;
  - Marca os tempos de descoberta e finalização de cada vértice;

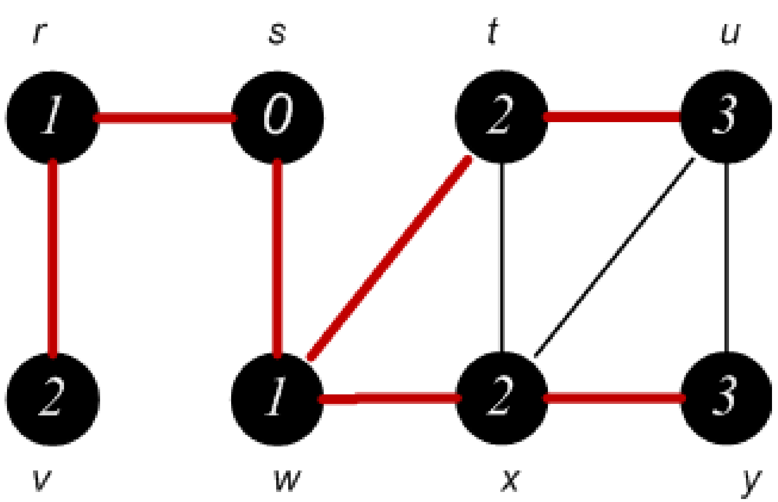
Busca em Largura

Busca em largura

- Um dos algoritmos mais simples da área de grafos;
- Serve de **base para** vários **outros algoritmos**:
  - Base para Caminho mais curto (*Dijkstra*);
    - Utilizado para calcular rotas de custo mínimo em um par de localidades em um mapa, por exemplo;
  - Base para Árvore Geradora Mínima - AGM (*Prim*);
    - Utilizado para interligar localidades a um custo mínimo, por exemplo.

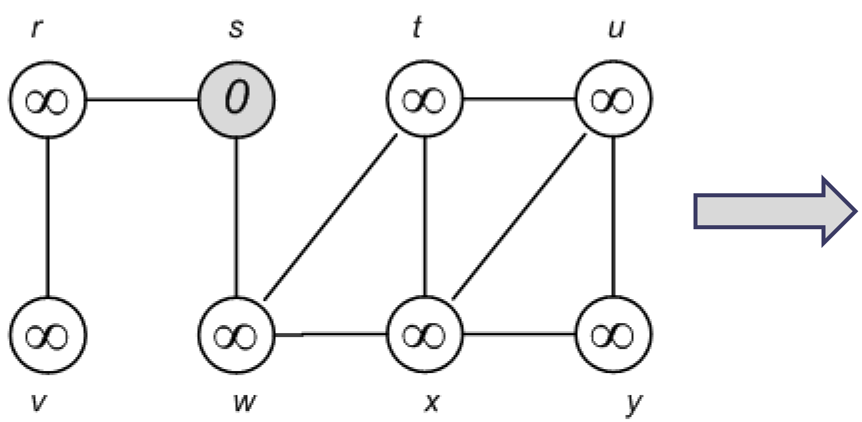
Busca em largura

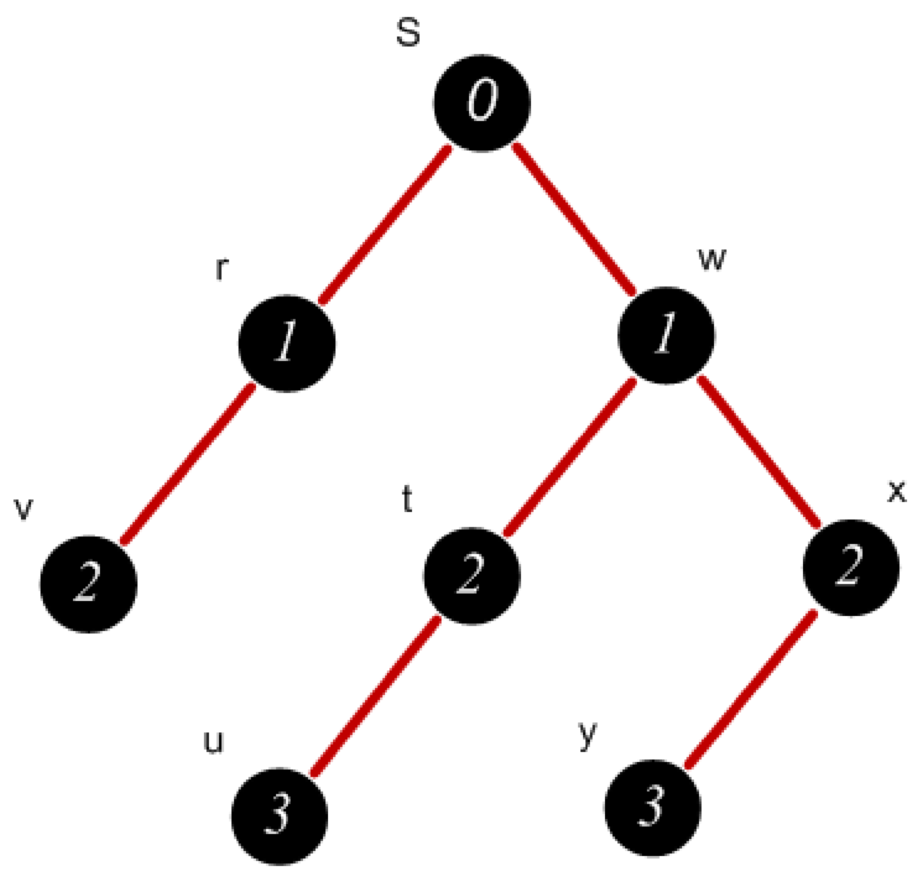
- O algoritmo da Busca em Largura calcula a distância (menor número de arestas) desde o vértice *s* (raiz) até todos os vértices acessíveis;
  - Não considera a distância como o somatório do peso de arestas;
  - Considera a quantidade de saltos necessários mínimos para alcançar outro vértice do grafo;



Busca em largura

- Ele também produz uma "Árvore Primeiro na Extensão", com raiz no vértice de partida, que contém todos os vértices acessíveis;



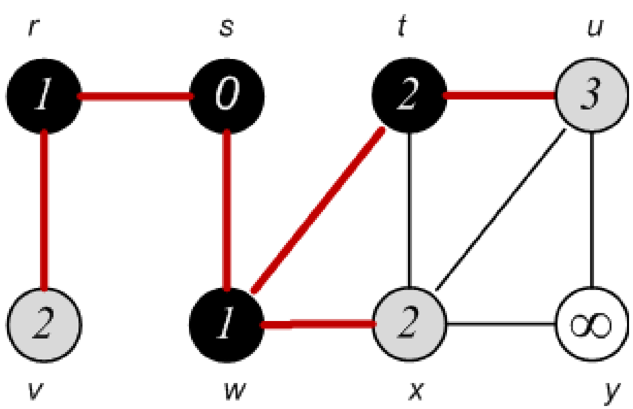


Busca em largura

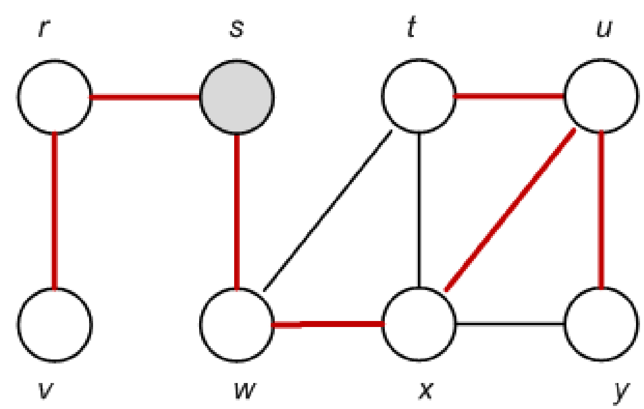
- Para cada vértice *v* acessível a partir de *s*, o caminho na árvore primeiro na extensão de *s* até *v* corresponde a um "caminho mais curto" de *s* até *v*, ou seja, um caminho que contém um número mínimo de arestas;
  - Só é possível porque a busca é "guiada de nível em nível";

- *Observação: Esta informação não é possível ser obtida na busca em profundidade*:

d(s,t)=2
na BFS

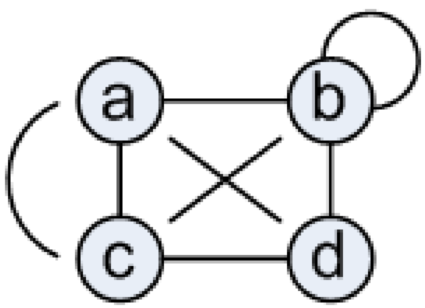


d(s,t)=4
na DFS

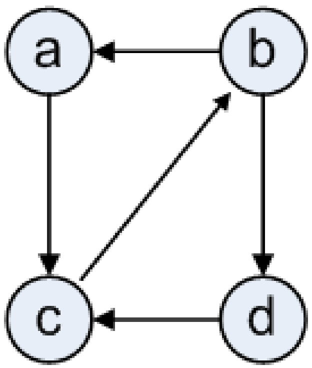


Busca em largura

- Assim como a *Busca em Profundidade (DFS)*, o algoritmo da *Busca em Largura (BFS)* funciona sobre grafos orientados e também não orientados;
  - O que importa, é a relação de adjacência;



G



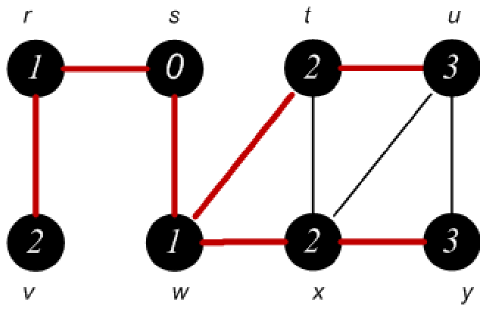
G'

Busca em largura

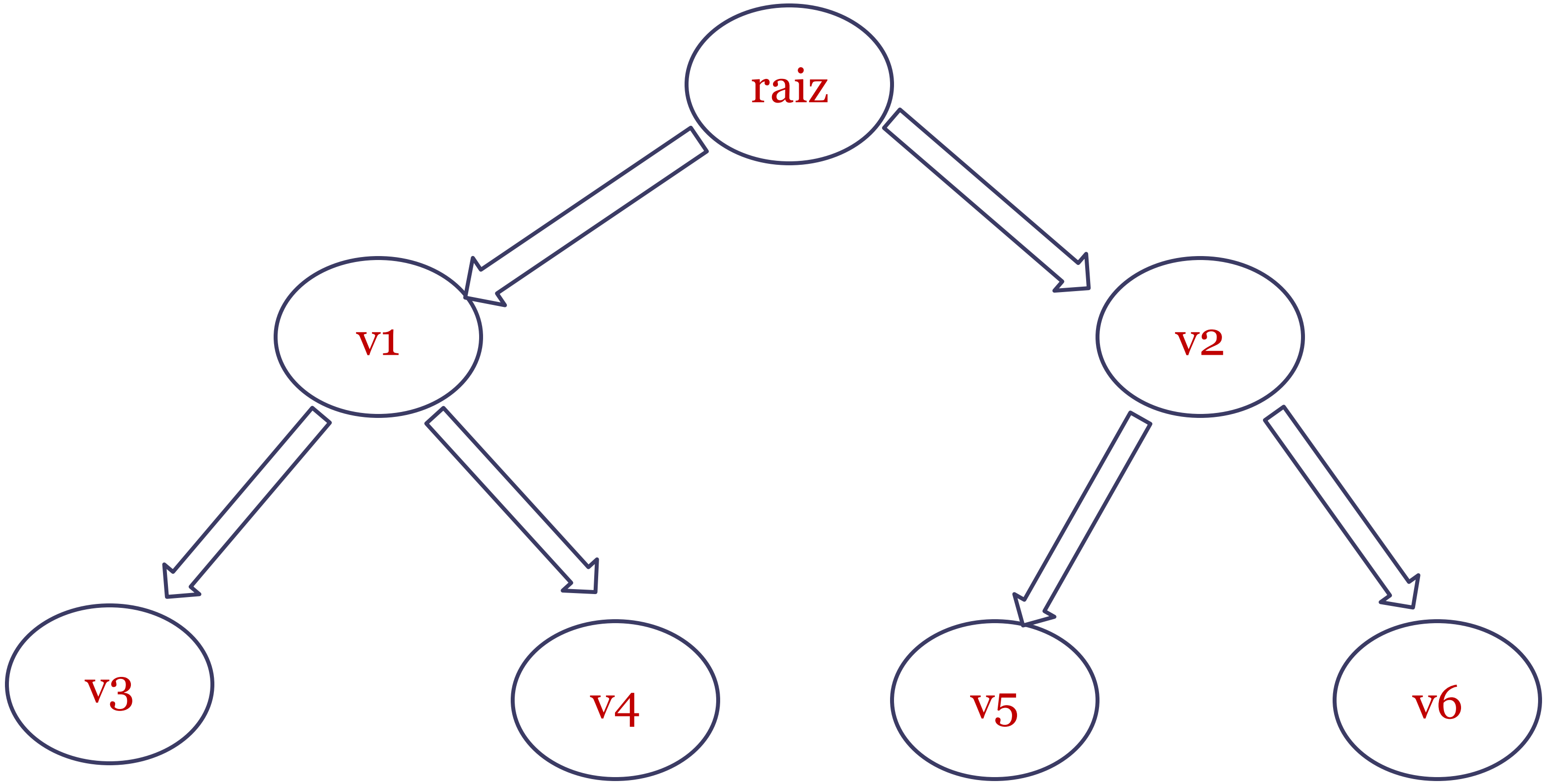
- A busca em largura recebe esse nome porque expande a fronteira entre vértices descobertos e não descobertos uniformemente ao longo da extensão da fronteira;

- Isto é, o algoritmo descobre todos os vértices à distância k a partir de s, antes de descobrir quaisquer vértices à distância $k+1$; *(ponto chave)*

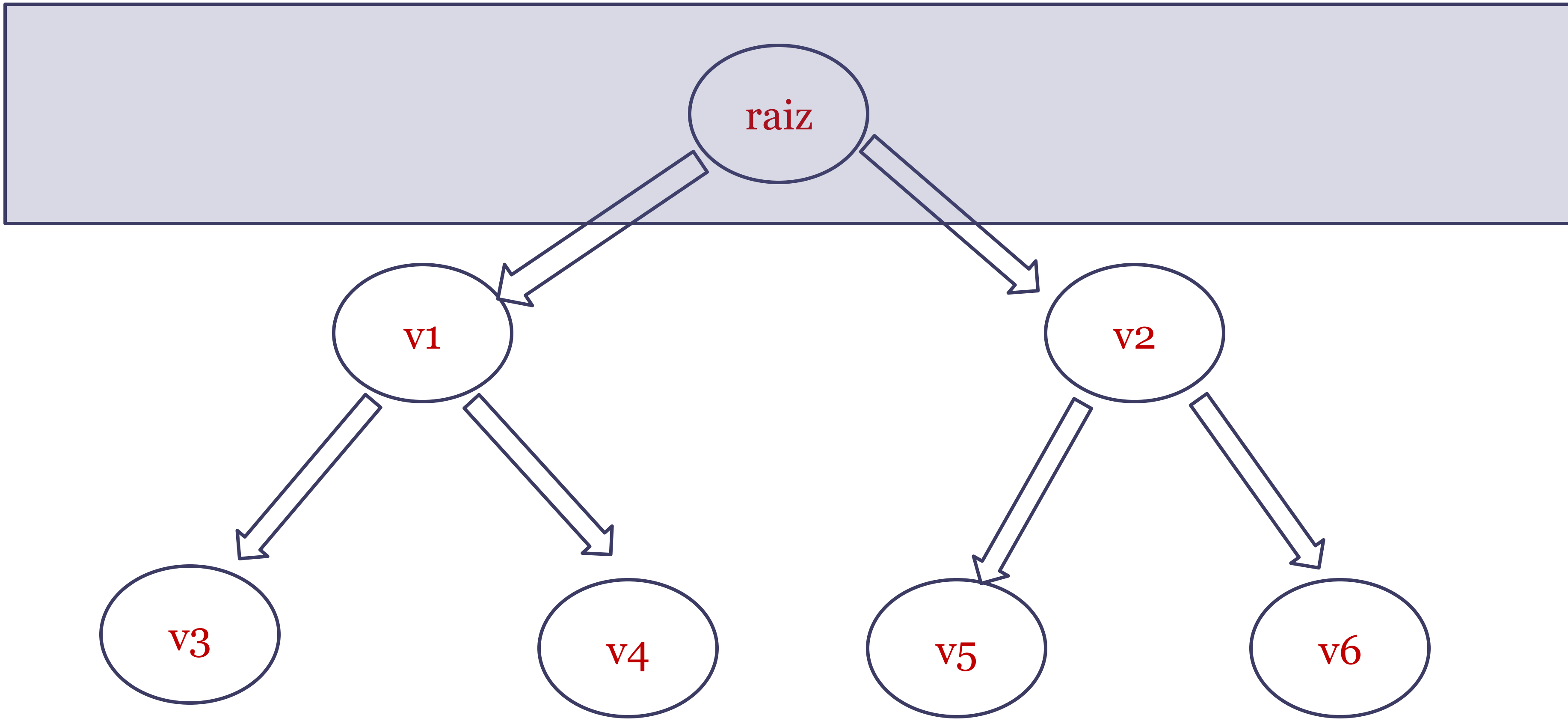
- Comparação com o movimento da água;



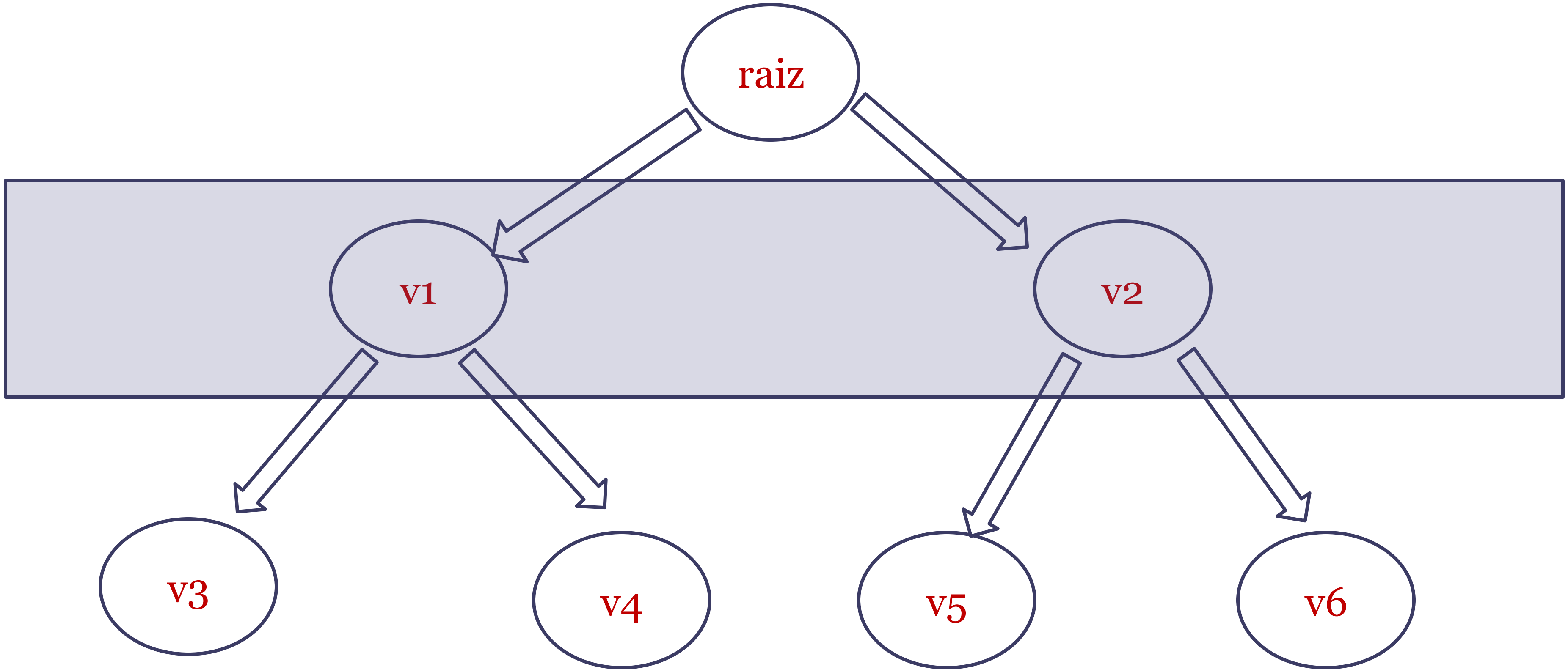
Aplicando Busca em Largura em uma Árvore



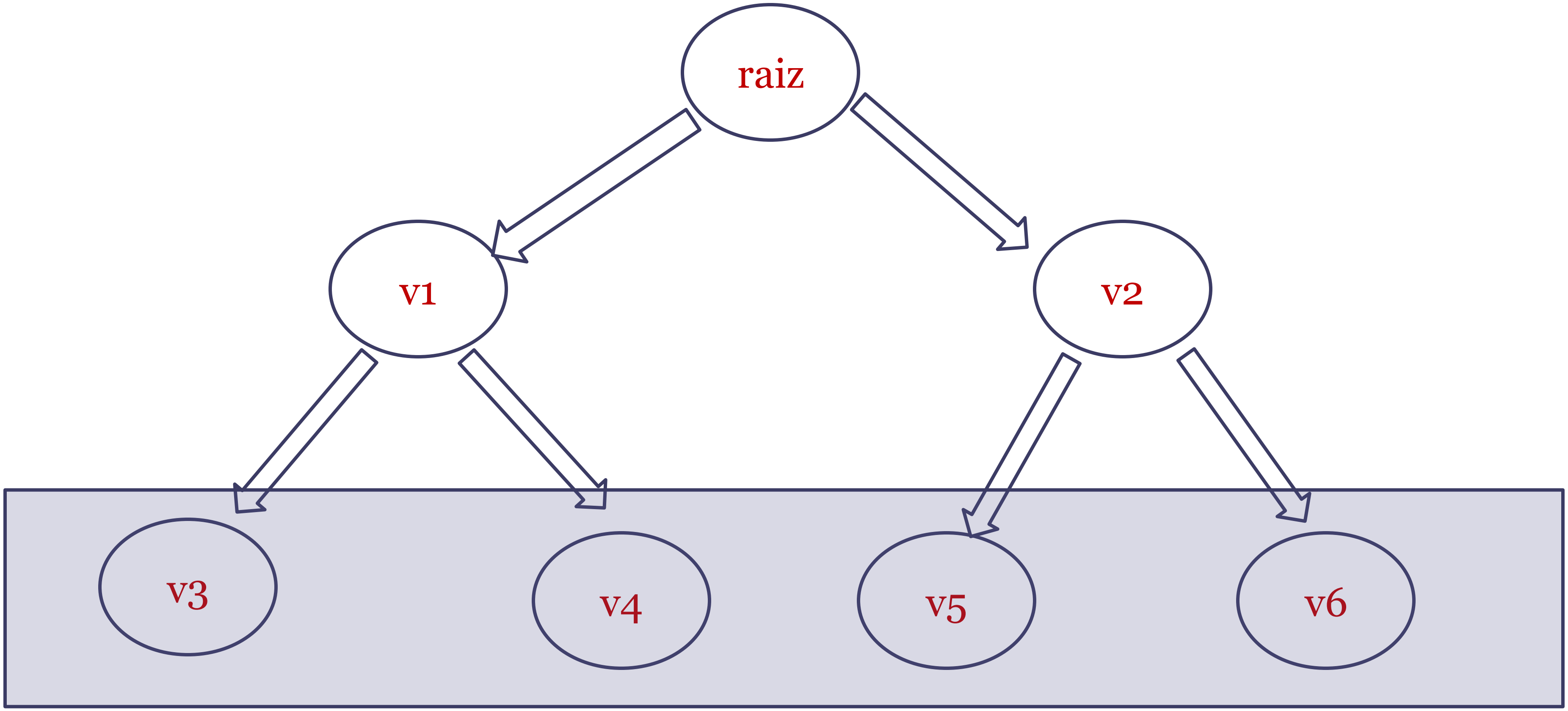
Aplicando Busca em Largura em uma Árvore



Aplicando Busca em Largura em uma Árvore



Aplicando Busca em Largura em uma Árvore



Busca em largura

- O controle do descobrimento dos nós na busca em largura é feito de forma semelhante ao controle utilizado na busca em profundidade anteriormente apresentada:
  - Nó branco = Não visitado/não conhecido;
  - Nó cinza = Nó conhecido/não visitado; Seus adjacentes não foram inseridos em uma fila;
  - Nó preto = Nó conhecido/Nó visitado; Todos os seus adjacentes foram inseridos na fila (não necessariamente visitados, como na DFS);

Busca em largura

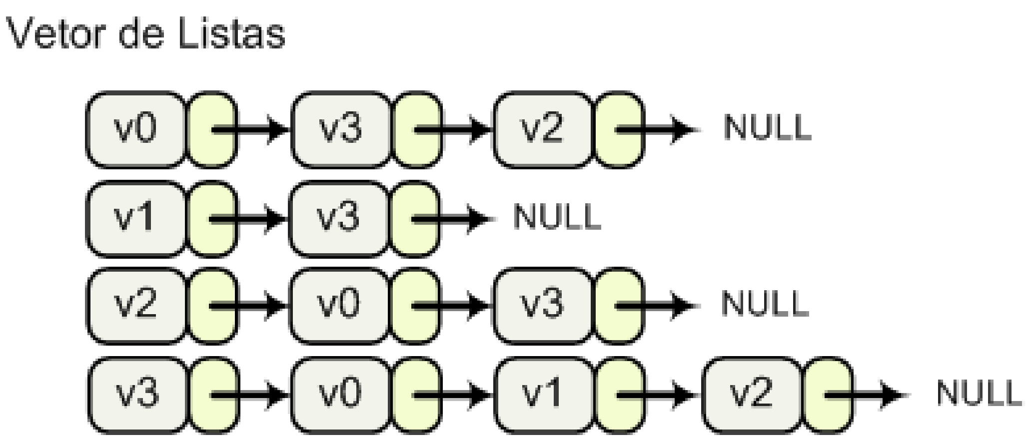
- Como um vértice é descoberto no máximo uma vez, este possui apenas um pai;
  - A relação de "pai" depende da organização em função da representação do grafo (especificamente da relação de adjacência);

- Conceito de Ancestral:
  - Se u está no caminho na árvore a partir da raiz s até o vértice v, então u é ancestral de v, e v é um descendente de u.

- Tudo depende do nó escolhido para raiz; *As vezes é prefixado, como em algumas aplicações da área de redes*;
  - Roteamento, por exemplo (montando tabelas de encaminhamento);

Busca em largura

- Segundo Cormen, a Busca em Largura (BFS) pressupõe que o grafo G=(V,A) é representado por uma lista de adjacência;
  - Mas isso não é uma total verdade na prática...



Busca em largura

- Assim como na DFS, a BFS faz uso de algumas estruturas auxiliares durante a pesquisa:
  - cor[u]; *//indicativo de atingibilidade;*
  - π[u]; *//indica o vértice predecessor de u(pai);*
  - d[u]; *//indica a distância desde a origem d(s,u) - em arestas;*
  - Q; *//indica a fila (FIFO) – ponto chave do algoritmo.*

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$cor[u] \leftarrow BRANCO$

$d[u] \leftarrow \infty$

$\pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\ Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$u \leftarrow DESENFILEIRA(Q)$

$para\ cada\ v \leftarrow Adj[u]$

$se\ cor[v] = BRANCO$

$cor[v] \leftarrow CINZA$

$d[v] = d[u] + 1$

$\pi[v] \leftarrow u$

$ENFILEIRA(Q, v)$

$cor[u] \leftarrow PRETO$

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

O procedimento BFS recebe como parâmetro o grafo G(V,A) e um vértice para iniciar a busca

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad\quad cada\ v \leftarrow Adj[u]$

CO

ZA

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Para cada vértice do grafo, diferente do vértice inicial s, faça...

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

Indica que eles estão descobertos (ainda não conhecidos)

BRANCOS

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\ Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad ... NCO$

$\quad\quad ... ZA$

$\quad\quad ... 1$

$\quad\quad ...$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Indica que a distância da Raiz *s* até cada vértice é infinita (a princípio)

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G]$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova Fila()$

$ENFILEIRA(Q, s)$

Indica que cada vértice ainda não tem predecessor/pai;

$ia(Q)$

$NFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\ Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$CINZA$

$+1$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

A cor do vértice de partida
s é Cinza... O primeiro a
ser conhecido/visitado...

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\ Fila()$

$ENFILEIRA(Q, s)$

distância(*s*, *s*) = 0
Óbvio!

$o\ !vazia(Q)$

$DESENFILEIRA(Q)$

$cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$
$\quad cor[u] \leftarrow BRANCO$
$\quad d[u] \leftarrow \infty$
$\quad \pi[u] \leftarrow NULL$
$cor[s] \leftarrow CINZA$
$d[s] \leftarrow 0$
$\pi[s] \leftarrow NULL$
$Q \leftarrow nova\ Fila()$
$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$
$\quad u \leftarrow DESENFILEIRA(Q)$
$\quad para\ cada\ v \leftarrow Adj[u]$
$\quad\quad se\ cor[v] = BRANCO$
$v] \leftarrow CINZA$
$= d[u] + 1$
$\leftarrow u$
$FILEIRA(Q, v)$
$\quad cor[u] \leftarrow PRETO$

E por *default*, o vértice de partida não possui predecessor (pai);

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$NFILEIRA(Q)$

$v \leftarrow Adj[u]$

$] = BRANCO$

$\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad d[v] = d[u] + 1$

$\quad\quad \pi[v] \leftarrow u$

$\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Uma estrutura auxiliar de fila é iniciada como vazia;

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$, v)$

E o vértice de partida *s* é enfileirado; O algoritmo está pronto para começar!!!

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[$

$\quad \pi$

$cor[$

$d[s]$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

Enquanto existir vértices ainda não visitados, faça...

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\qquad se\ cor[v] = BRANCO$

$\qquad\quad cor[v] \leftarrow CINZA$

$\qquad\quad d[v] = d[u] + 1$

$\qquad\quad \pi[v] \leftarrow u$

$\qquad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[$

$\quad \pi$

$cor[$

$d[s]$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Retira o primeiro da fila (u)...

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[$

$\quad \pi$

$cor[$

$d[s]$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\ Fila()$

$ENFILEIRA(Q, s)$

Para todos os adjacentes de u, faça..

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[$

$\quad \pi$

$cor[$

$d[s]$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

Se o adjacente de *u* ainda não é conhecido, faça...

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em Largura

```
BFS(G, s)
para cada vértice u ← V[G] − {s}
    cor[u] ← BRANCO
    d[
    π
cor[
d[s]
π[s] ← NULL
Q ← novaFila()
ENFILEIRA(Q, s)
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u] + 1
            π[v] ← u
            ENFILEIRA(Q, v)
    cor[u] ← PRETO
```

Colora-o de CINZA...
(conhecido/não visitado)

Busca em Largura

```
BFS(G, s)
para cada vértice u ← V[G] − {s}
    cor[u] ← BRANCO
    d[u] ← ∞
    π[u] ← NULL
cor[s] ← CINZA
d[s] ← 0
π[s] ← NULL
Q ← no
ENFILI
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u] + 1
            π[v] ← u
            ENFILEIRA(Q, v)
    cor[u] ← PRETO
```

Indica que a distância de v até a raiz é uma unidade a mais que a distância de seu pai até a raiz...

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow$

$Q \leftarrow no$

$ENFIL$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Seta o pai do vértice v como u.

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u$

$cor[s$

$d[s] \leftarrow$

$\pi[s] \leftarrow$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Enfileira o vértice *v* para explorar seus adjacentes no futuro...

Busca em Largura

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s]$ …

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Ao conhecer todos os adjacentes de u, o vértice é marcado como PRETO. (conhecido/visitado)

Busca em Largura

$BFS(G, s)$

$para$

co

$d[$

$\pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$u \leftarrow DESENFILEIRA(Q)$

$para\ cada\ v \leftarrow Adj[u]$

$se\ cor[v] = BRANCO$

$cor[v] \leftarrow CINZA$

$d[v] = d[u] + 1$

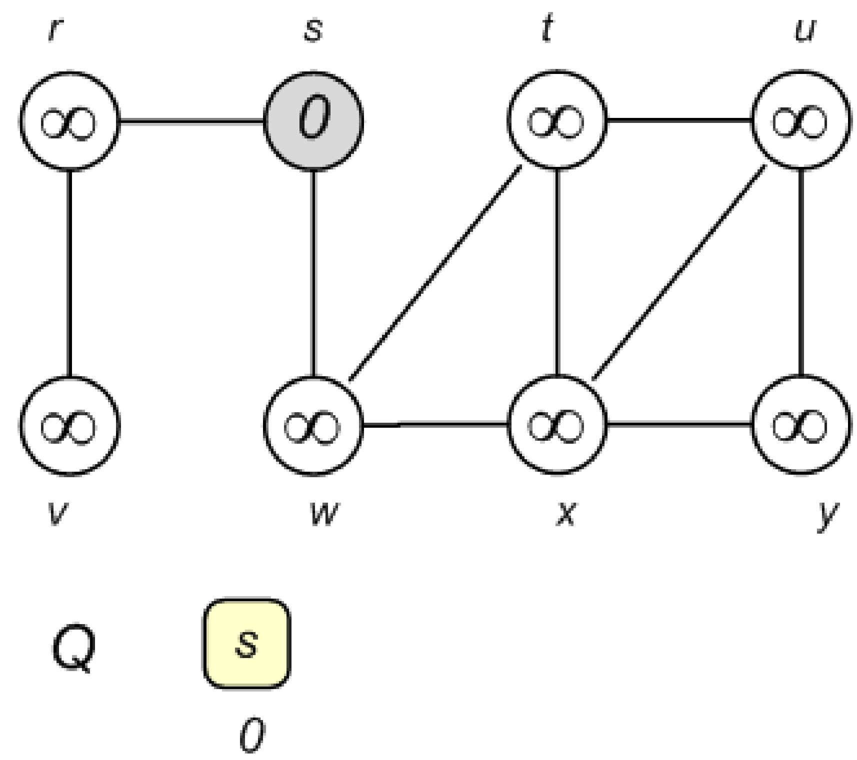
$\pi[v] \leftarrow u$

$ENFILEIRA(Q, v)$

$cor[u] \leftarrow PRETO$

Lembrando que este procedimento se repete até que a fila esteja vazia...

Busca em Largura



$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

Inicializa as variáveis da BFS

Busca em Largura

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

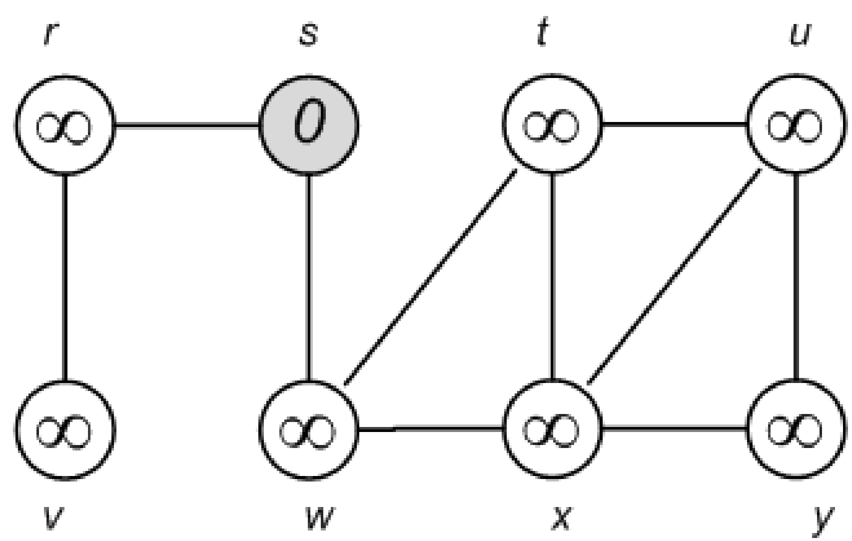
$\quad\quad\quad cor[v] \leftarrow CINZA$

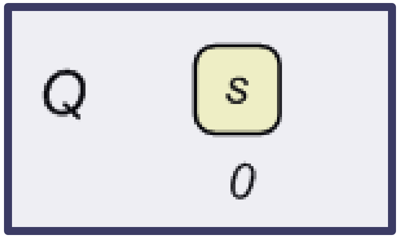
$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

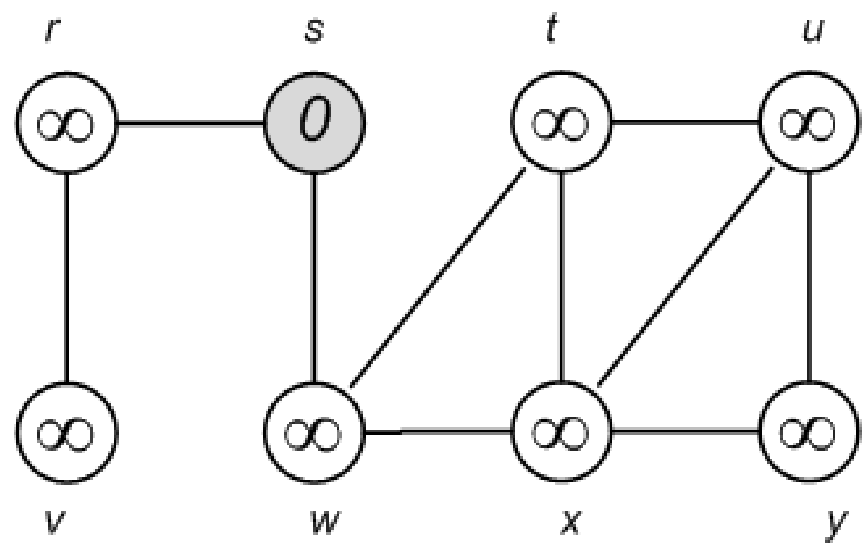
$\quad cor[u] \leftarrow PRETO$





A fila não está vazia!

Busca em Largura



$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\qquad se\ cor[v] = BRANCO$

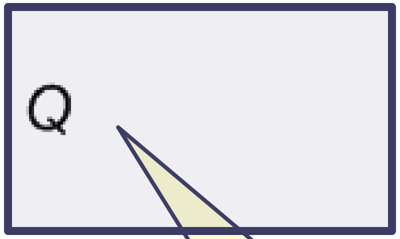
$\qquad\quad cor[v] \leftarrow CINZA$

$\qquad\quad d[v] = d[u]+1$

$\qquad\quad \pi[v] \leftarrow u$

$\qquad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

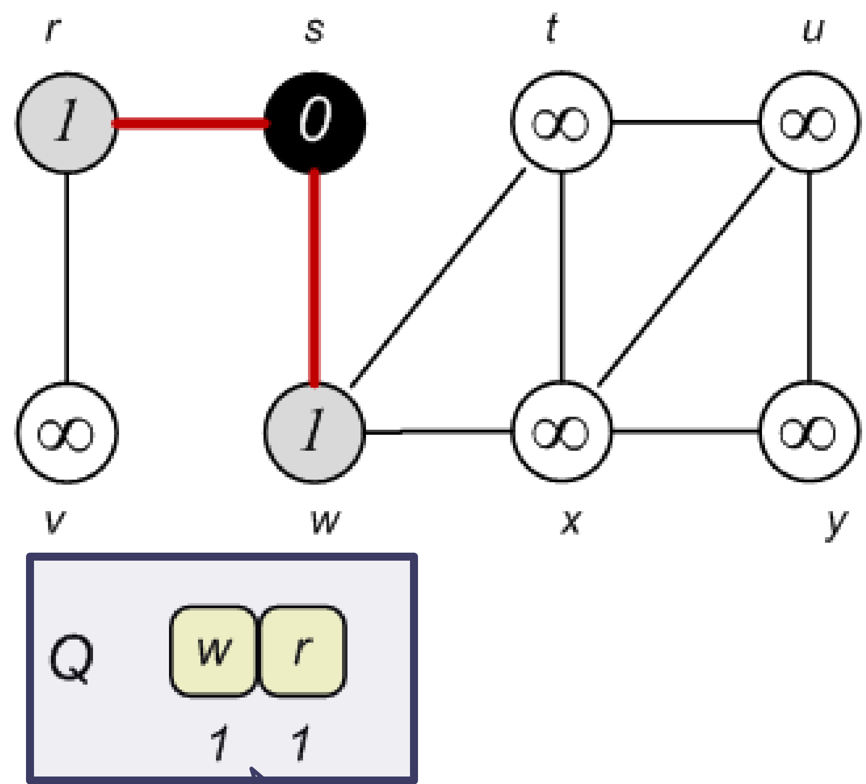


u=s

Adj[u]={r,w}

Retira *s* da fila, e parte para seus adjacentes...

Busca em Largura



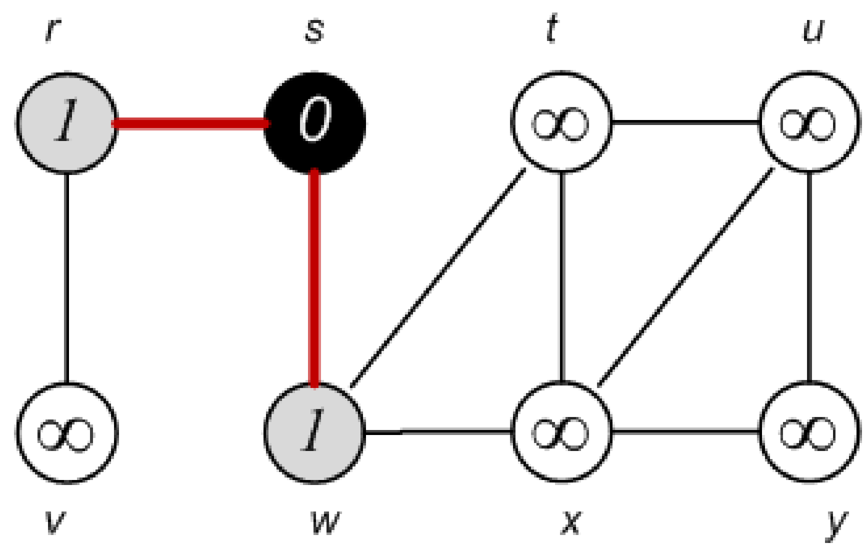

```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q, v)
    cor[u] ← PRETO
```

u=s

Adj[u]={r,w}

Enfileirou os vértices desconhecidos pela busca...

Busca em Largura



enquanto !*vazia*(*Q*)

$u \leftarrow DESENFILEIRA(Q)$

para cada $v \leftarrow Adj[u]$

se $cor[v] = BRANCO$

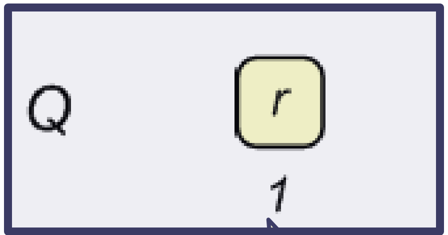
$cor[v] \leftarrow CINZA$

$d[v] = d[u] + 1$

$\pi[v] \leftarrow u$

$ENFILEIRA(Q, v)$

$cor[u] \leftarrow PRETO$



u=w
Adj[u]={s,t,x}

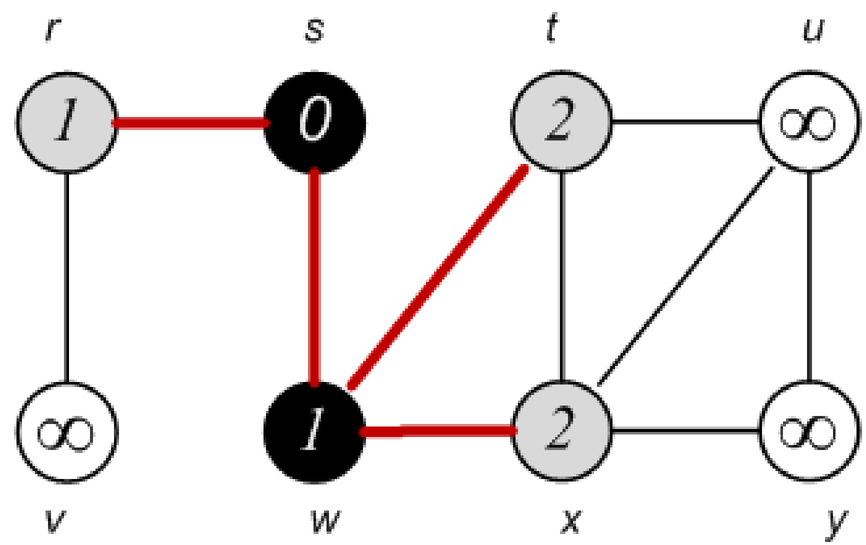
Retira *w* da fila, testa seus adjacentes...

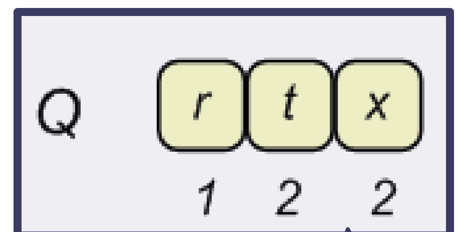
Busca em Largura

$$
\begin{array}{l}
\textit{enquanto } !\textit{vazia}(Q) \\
\quad u \leftarrow \textit{DESENFILEIRA}(Q) \\
\quad \textit{para cada } v \leftarrow \textit{Adj}[u] \\
\quad\quad \textit{se } cor[v] = \textit{BRANCO} \\
\quad\quad\quad cor[v] \leftarrow \textit{CINZA} \\
\quad\quad\quad d[v] = d[u] + 1 \\
\quad\quad\quad \pi[v] \leftarrow u \\
\quad\quad\quad \textit{ENFILEIRA}(Q, v) \\
\quad cor[u] \leftarrow \textit{PRETO}
\end{array}
$$

u=s

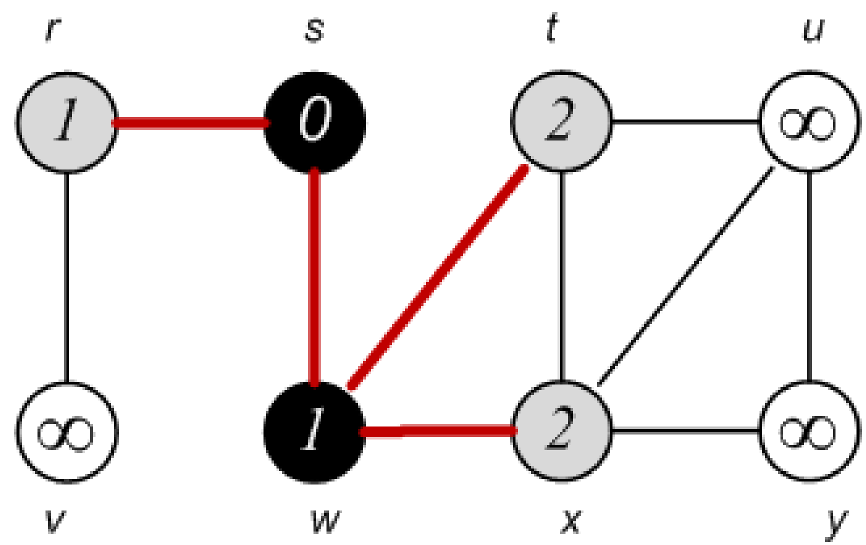
Adj[u]={t,x}



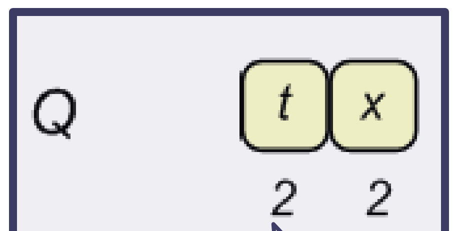


Enfileirou os vértices desconhecidos pela busca...

Busca em Largura



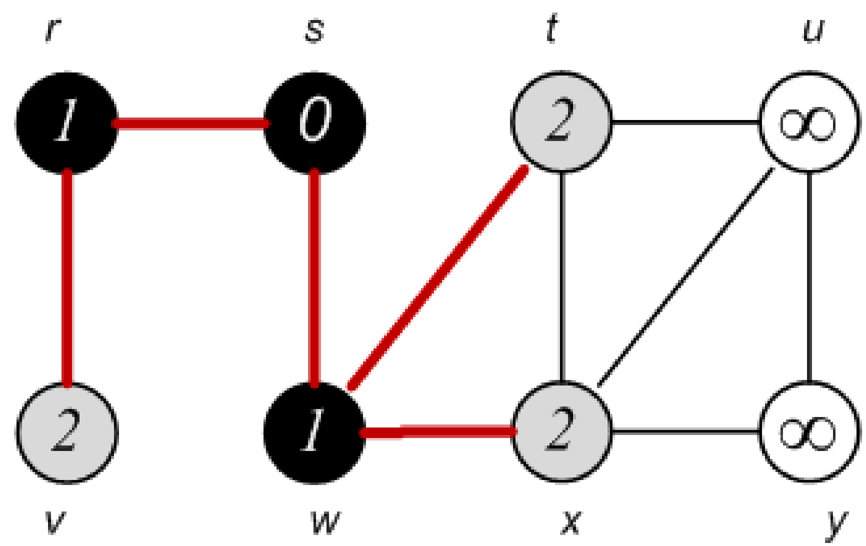

```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q, v)
    cor[u] ← PRETO
```



u=r
Adj[u]={s,v}

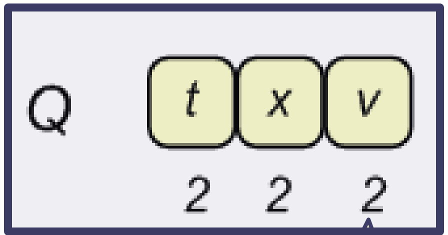
Retira *r* da fila, testa seus adjacentes...

Busca em Largura



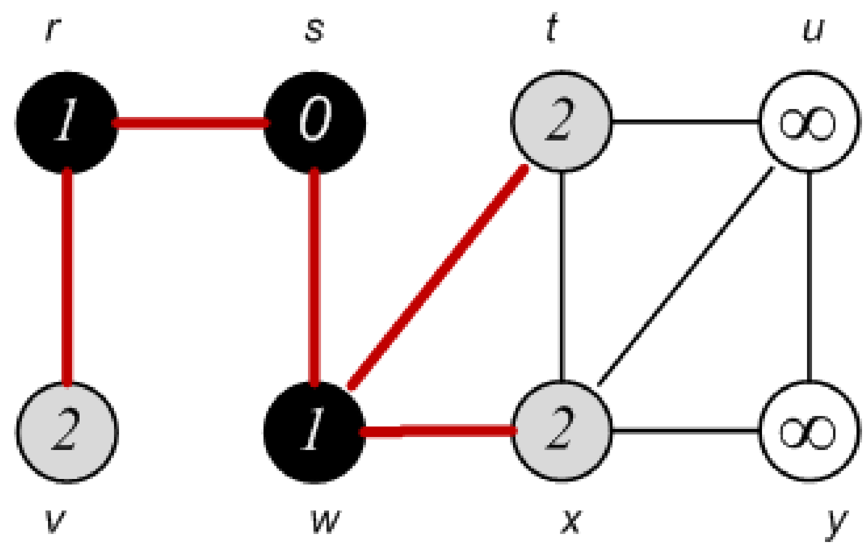

```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q, v)
    cor[u] ← PRETO
```

u=r
Adj[u]={s,v}

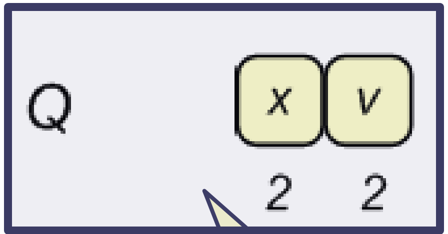


Enfileirou os vértices desconhecidos pela busca...

Busca em Largura




```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q,v)
    cor[u] ← PRETO
```

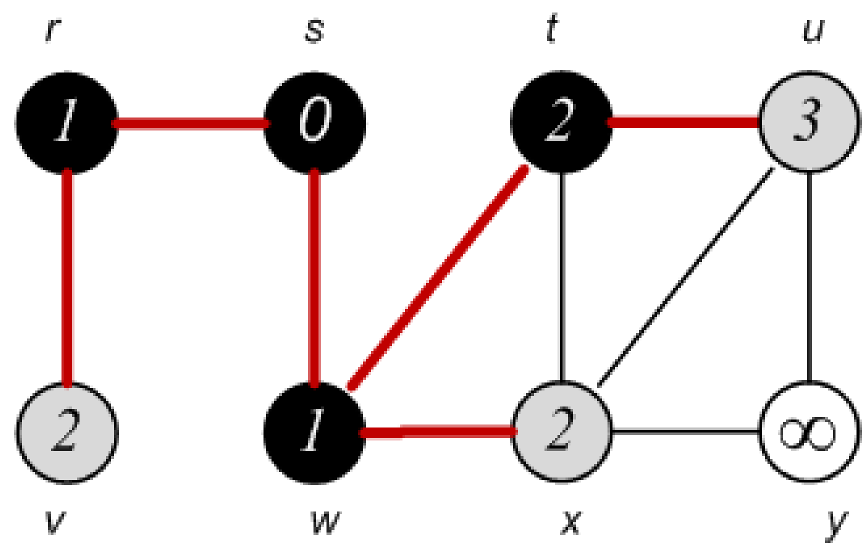


u=t

Adj[u]={w,x,u}

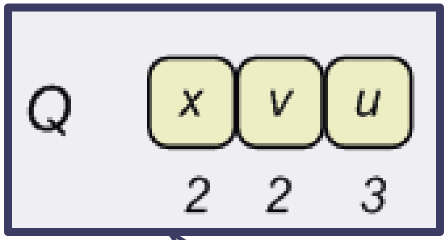
Retira *t* da fila, testa seus adjacentes...

Busca em Largura




```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q, v)
    cor[u] ← PRETO
```

u=t
Adj[u]={w,x,u}



Enfileirou os vértices desconhecidos pela busca...

Busca em Largura

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

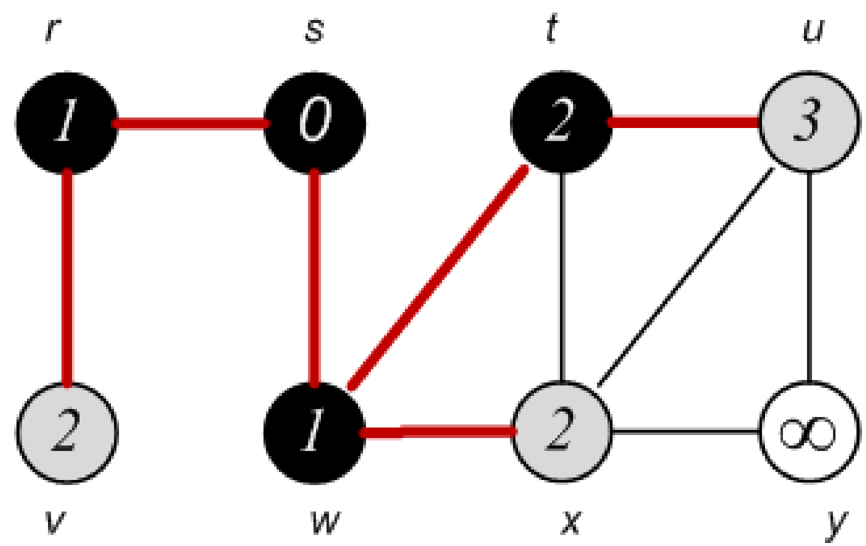
$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u]+1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

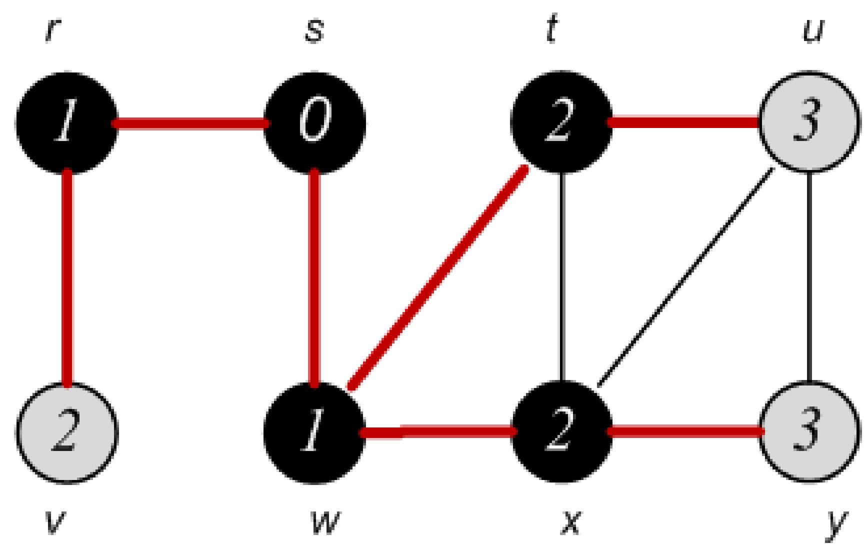


Retira *x* da fila, testa seus adjacentes...

u=x

Adj[u]={w,t,u,y}

Busca em Largura



$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u]+1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

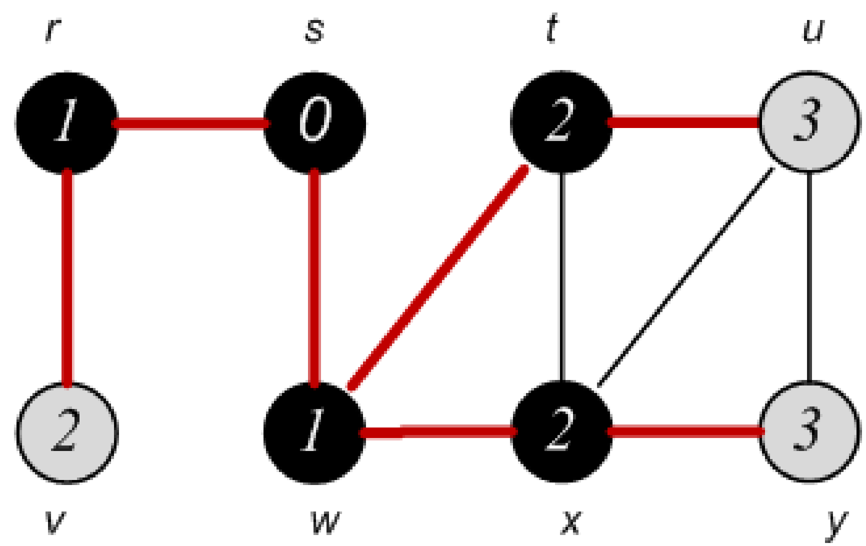
$\quad cor[u] \leftarrow PRETO$

Enfileirou os vértices desconhecidos pela busca...

u=x

Adj[u]={w,t,u,y}

Busca em Largura



$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

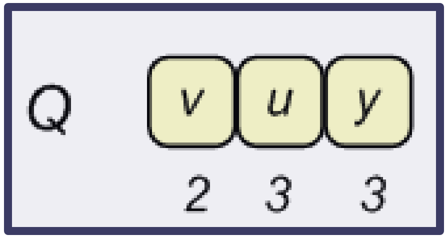
$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

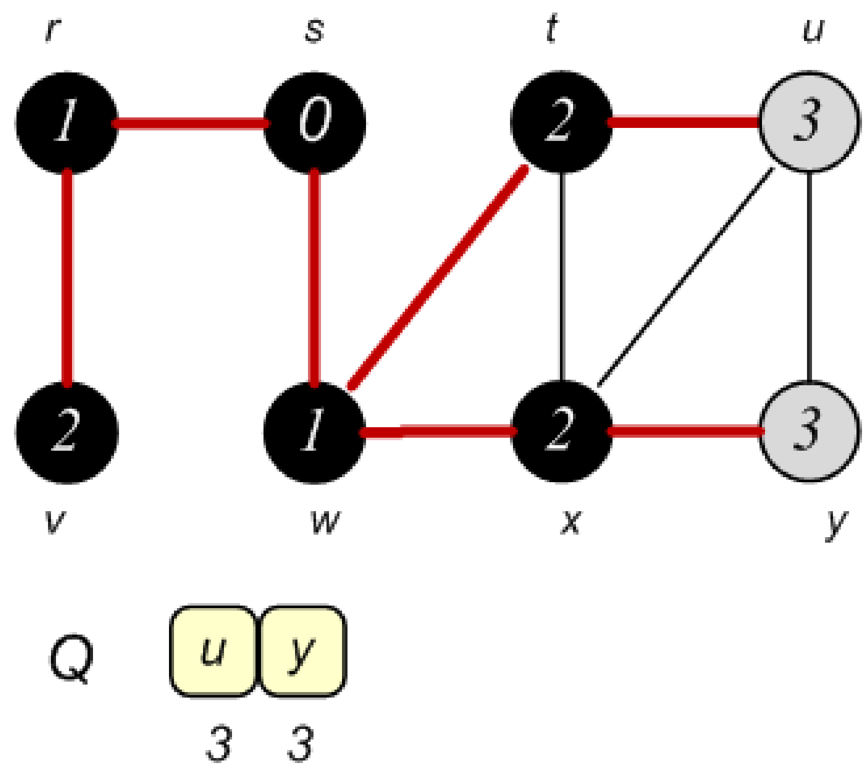
$\quad cor[u] \leftarrow PRETO$



Marcando os vértices de preto...

Busca em Largura

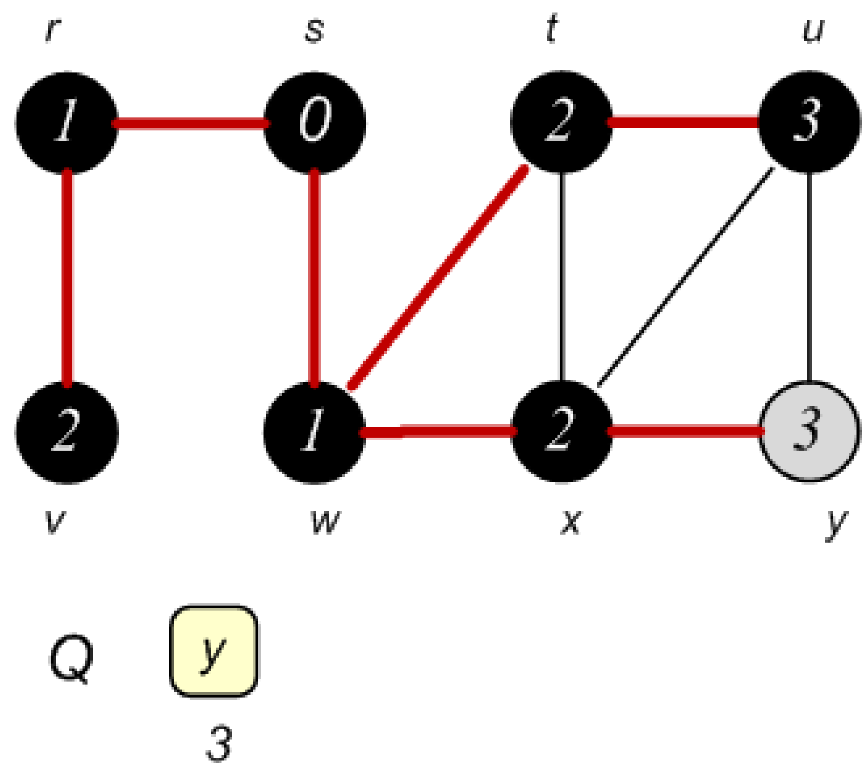
```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q,v)
    cor[u] ← PRETO
```



Marcando os vértices de preto...

Busca em Largura

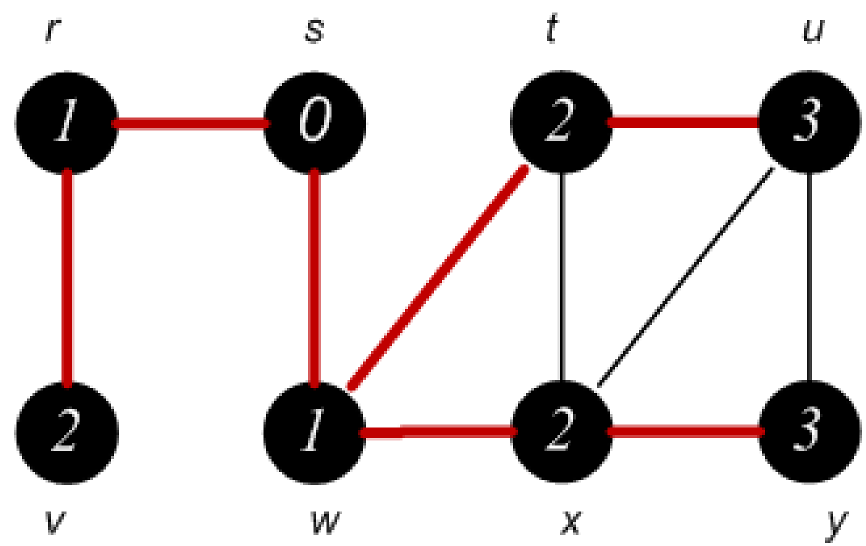
```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q,v)
    cor[u] ← PRETO
```



Marcando os vértices de preto...

Busca em Largura

```
enquanto !vazia(Q)
    u ← DESENFILEIRA(Q)
    para cada v ← Adj[u]
        se cor[v] = BRANCO
            cor[v] ← CINZA
            d[v] = d[u]+1
            π[v] ← u
            ENFILEIRA(Q,v)
    cor[u] ← PRETO
```

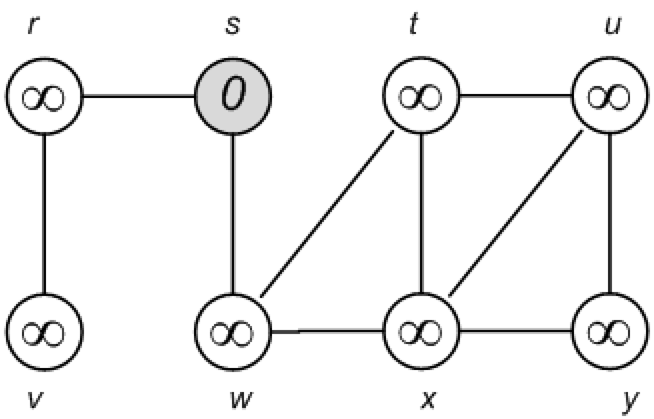


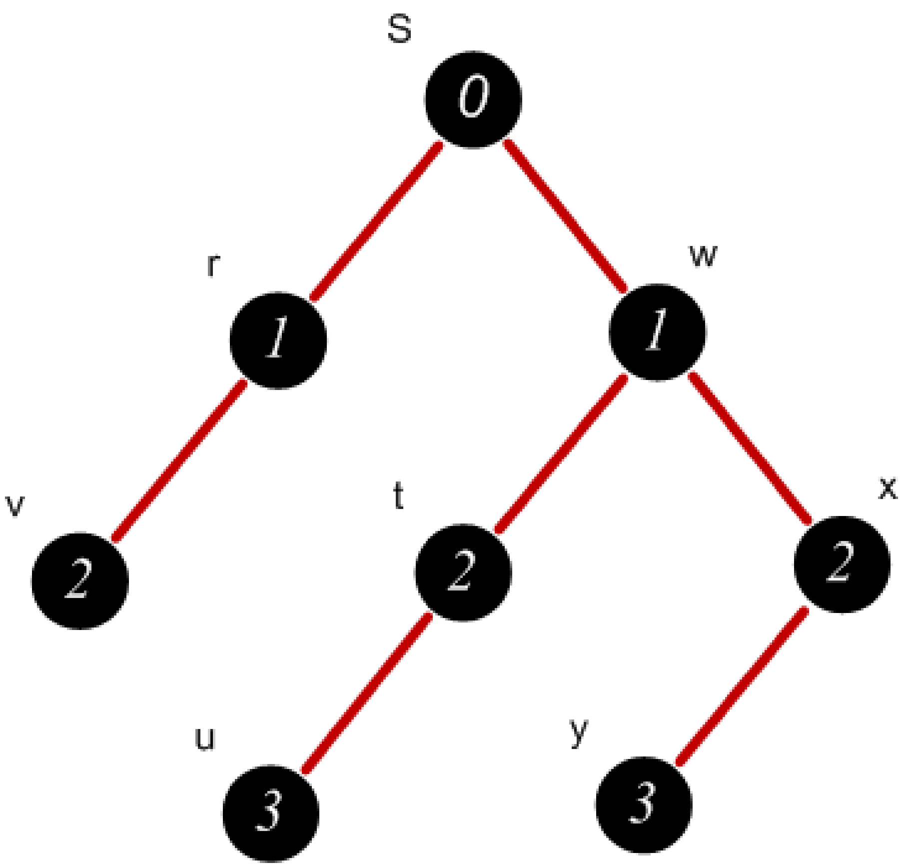
Q

Marcando os vértices de preto...

Busca em Largura

Árvore gerada na busca





Vetor Π

Índice:	s	y	x	t	w	u	v	r
Valor:	NULL	x	w	w	s	t	r	s

Busca em largura

Análise de complexidade

$BFS(G, s)$

$para\ cada\ vértice\ u \leftarrow V[G] - \{s\}$

$\quad cor[u] \leftarrow BRANCO$

$\quad d[u] \leftarrow \infty$

$\quad \pi[u] \leftarrow NULL$

$cor[s] \leftarrow CINZA$

$d[s] \leftarrow 0$

$\pi[s] \leftarrow NULL$

$Q \leftarrow nova\,Fila()$

$ENFILEIRA(Q, s)$

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\quad\quad se\ cor[v] = BRANCO$

$\quad\quad\quad cor[v] \leftarrow CINZA$

$\quad\quad\quad d[v] = d[u] + 1$

$\quad\quad\quad \pi[v] \leftarrow u$

$\quad\quad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

Busca em largura

Análise de complexidade

- Obviamente que a complexidade da busca em largura depende diretamente da representação do grafo utilizada;

$enquanto\ !vazia(Q)$

$\quad u \leftarrow DESENFILEIRA(Q)$

$\quad para\ cada\ v \leftarrow Adj[u]$

$\qquad se\ cor[v] = BRANCO$

$\qquad\quad cor[v] \leftarrow CINZA$

$\qquad\quad d[v] = d[u] + 1$

$\qquad\quad \pi[v] \leftarrow u$

$\qquad\quad ENFILEIRA(Q, v)$

$\quad cor[u] \leftarrow PRETO$

- Utilizando lista de adjacência: $O(|V| + |A|)$

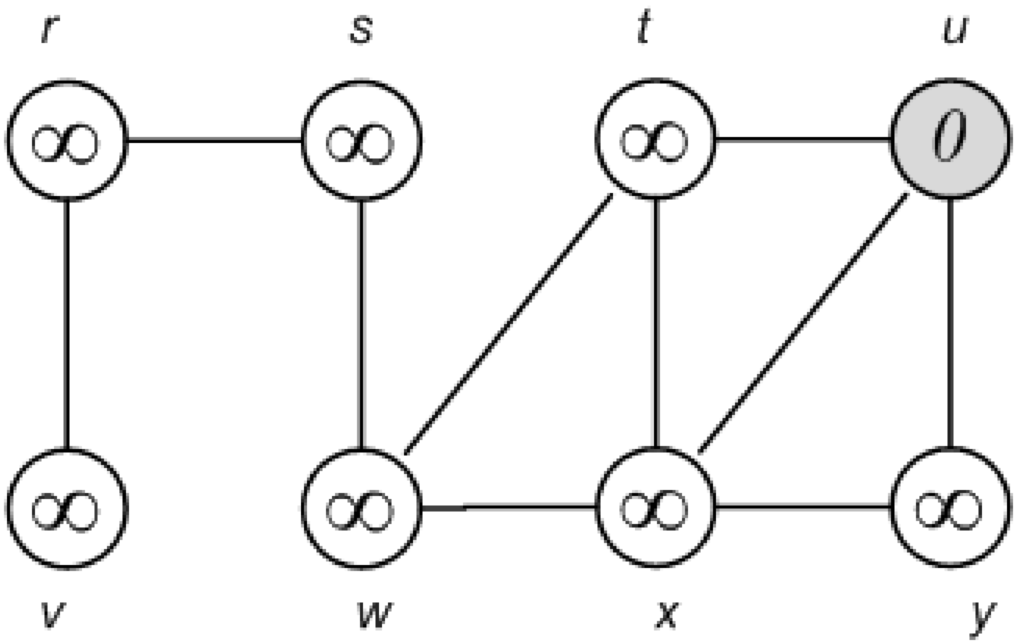
Exercícios

Exercício 01

- Sugira adaptações simples no algoritmo de Busca em Largura para transformá-lo em uma Busca em profundidade;
  - Apresente o pseudo-código;
  - Apresente também uma discussão sobre sua solução.

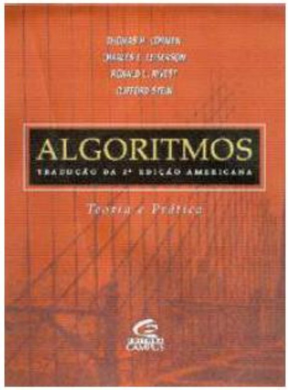
Exercício 02

- Mostre os valores dos vetores π e d resultantes da *BFS(G,u)*, para o grafo G a seguir:



Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos – Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus.



- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson;